

3 Praktikum: Rot-Schwarz-Baum

Sie sind nach wie vor als Systemarchitekt bei Data-Fuse und konnten Ihre Kollegen bereits mit den Programm zur Datenhaltung und -verarbeitung überzeugen. Es hat sich aber herausgestellt, dass ein Binärbaum zur Datenhaltung hier nicht optimal ist. Bei großen Datenmengen kann es zu entarteten Bäumen kommen. Um z.B. die Suche zu optimieren, sollen Sie dies durch die Organisation mittels eines ausgeglichenen Baumes ersetzen. Hierzu wählen Sie einen Rot-Schwarz-Baum mit der Top-Down Einfügemethode.

3.1 Aufgabenstellung

Übernehmen Sie Baum- und Knotenklasse aus dem 2. Praktikum und erweitern Sie die Klassen wie in den UML Diagrammen angegeben (neue bzw entfernte Attribute/Methoden in rot markiert).

Beachten Sie ALLE Lösungshinweise und setzen Sie sich zuerst mit der Funktionsweise eines Rot-Schwarz-Baum auseinander.

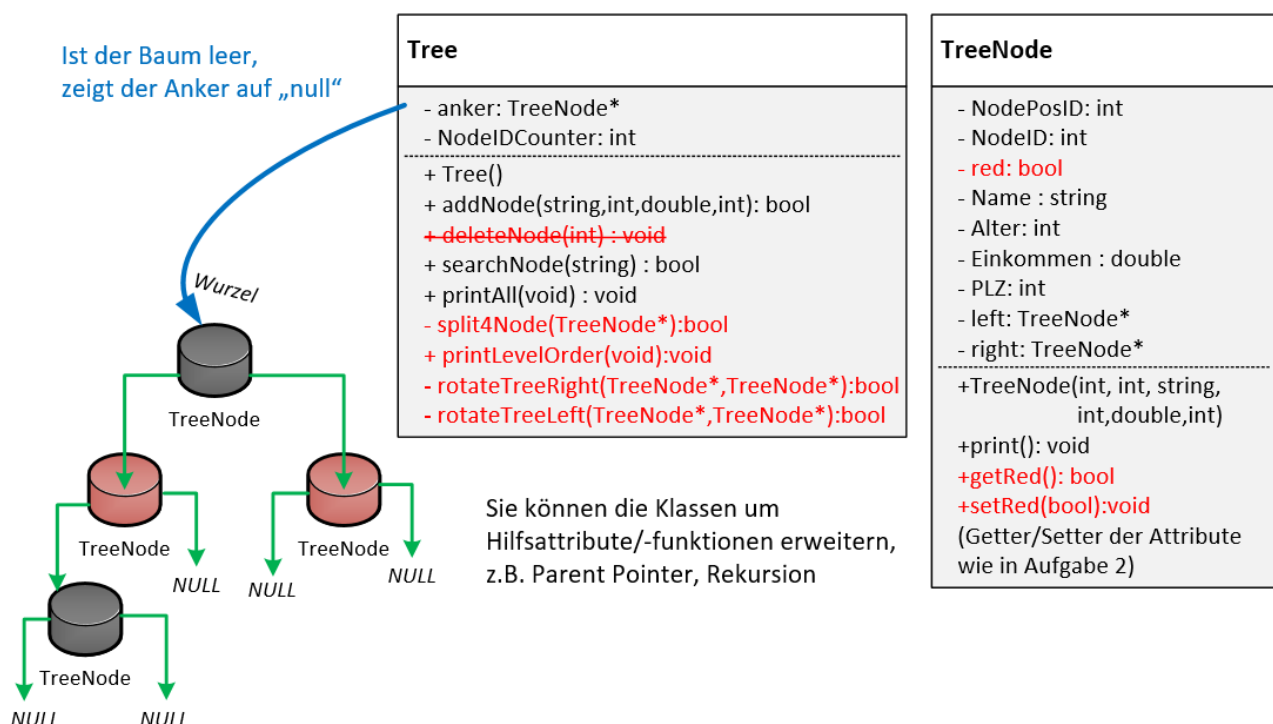


Abbildung 1: Aufbau der Baumstruktur und Erweiterung der Klassen.

In der Vorlesung wurde zunächst der 2-3-4-Baum vorgestellt. Das ist ein geordneter Suchbaum, bei dem die Knoten einen, zwei oder drei Schlüssel aufnehmen können und entsprechend zwei, drei oder vier Nachfolger haben können. Die Implementierung eines 2-3-4-Baumes ist mit einem Rot-Schwarz-Baum möglich. So lässt sich ein gültiger Rot-Schwarz-Baum jederzeit in einen 2-3-4-Baum überführen und umgekehrt. Eine Hauptaufgabe in diesem Praktikum ist das Einfügen eines neuen Knotens in einen Rot-Schwarz-Baum. Die Abbildungen 2 bis 7 zeigen die einzelnen Schritte beim Einfügen eines Knotens 26 in den gegebenen Rot-Schwarz-Baum. Im Endergebnis sieht man, dass der Rot-Schwarz-Baum eine gültige Schwarzausgeglichenheit hat, dh. die schwarzen Kanten entsprechen den Referenzen im 2-3-4-Baum (s. Abb. reffig:RS7).

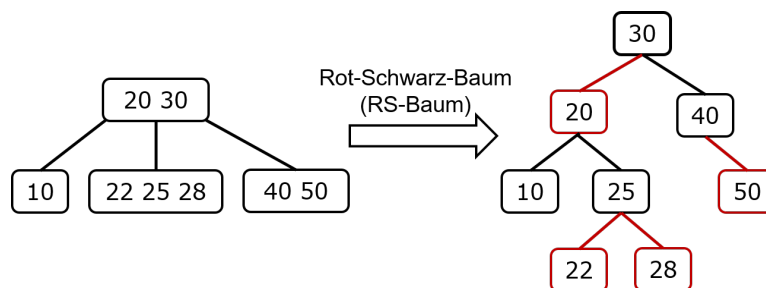


Abbildung 2: 2-3-4-Baum als Rot-Schwarz-Baum.

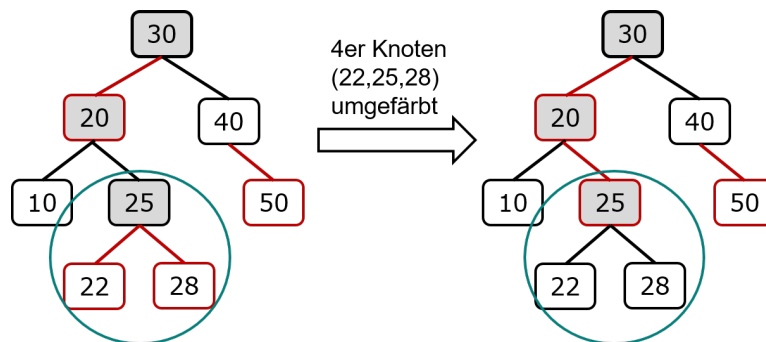


Abbildung 3: Umwandlung eines 4er-Knotens in zwei 2er Knoten entspricht einer Umfärbung eines schwarzen Knotens mit 2 roten Nachfolgern im RS-Baum.

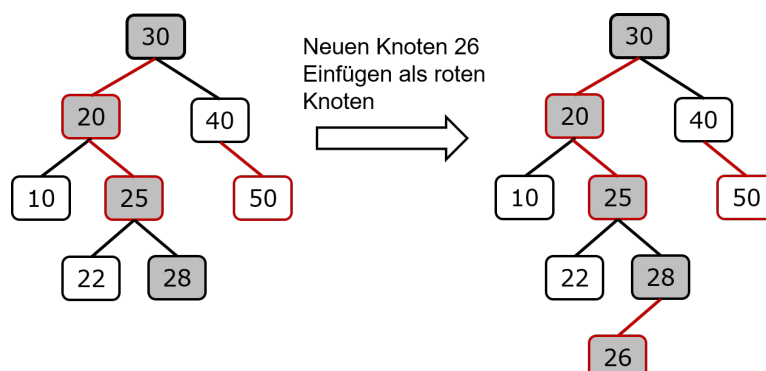


Abbildung 4: Einfügen des neuen Knotens 26 als roten Blattknoten. Der Suchpfad bis zum Knoten 28 bzw. 26 ist grau markiert.

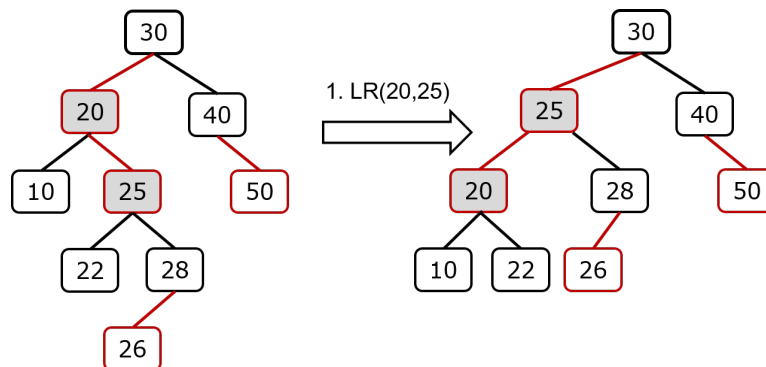


Abbildung 5: Bottom-Up Verfolgung des Suchpfades zur 26: Detektion von 2 roten Knoten. Da diese hier im Links-Rechts-Weg(Knick) erfolgen, werden die Rotationen Bottom-Up durchgeführt. Hier: Die erste Rotation ist eine Links-Rotation zwischen den Knoten 20 und 25.

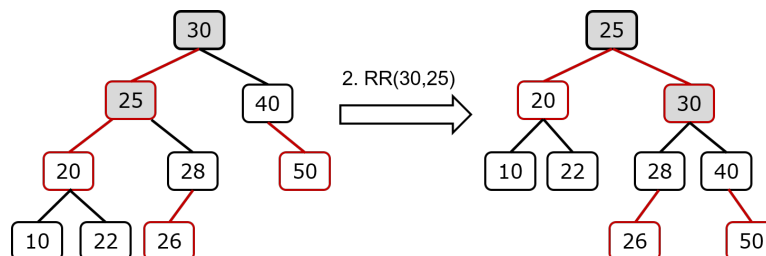


Abbildung 6: Durchführung der zweiten Rotation. Hier: Rechts-Rotation zwischen den Knoten 30 und 25.

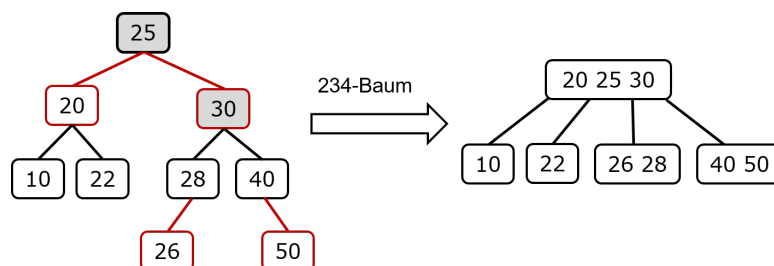


Abbildung 7: Ergebnis nach dem Einfügen des neuen Knoten 26 als Rot-Schwarz-Baum und als 2-3-4-Baum.

Die zu implementierenden Funktionen übernehmen dabei folgende Aufgaben:

- **bool split4Node(TreeNode*)**

Diese Methode soll in einem Rot-Schwarz-Baum einen Knoten überprüfen, ob dieser ein 4er Knoten ist. Ein 4er Knoten im 234-Baumes entspricht im Rot-Schwarz-Baum ein schwarzer Knoten, der zwei rote Knoten als Nachfolger hat. Dieser 4er Knoten soll im Sinne des 234-Baumes in zwei 2-er Knoten umgewandelt werden. Das bedeutet im Rot-Schwarz-Baum, dass die drei beteiligten Knoten nur umgefärbt werden müssen. Der schwarze Knoten wird dann rot und die vorab roten Knoten werden dann schwarz (s. Abb. 3).

Der Eingabeparameter dieser Methode sollte ein schwarzer Knoten sein. Die Methode soll generell erst überprüfen, ob es sich um einen 4er Knoten handelt und `true` zurück geben, wenn das der Fall ist und die Farben umgefärbt wurden.

- **bool addNode(string, int, double,int)**

Schreiben Sie eine **iterative Methode**, die einen neuen Knoten **TopDown** in dem Rot-Schwarz-Baum einfügt. Die Abbildungen 2 bis 7 zeigen Ihnen die Funktionsweise. Dabei ist folgenderweise vorzugehen:

1. Traversieren Sie den Baum von der Wurzel bis zum Blatt entsprechend dem einzufügenden Wert. Die Knoten des Suchpfades, die sich im Beispiel beim Einfügen der 26 ergeben, sind in der Abbildung 4 grau markiert. Wird auf dem Suchpfad von der Wurzel bis zum Blatt ein 4er-Knoten erkannt, wird dieser auf dem Weg nach unten (TopDown) mit der Methode `split4Node(TreeNode*)` umgefärbt. Merken Sie sich die Knoten des Suchpfades in einer weiteren Datenstruktur (z.Bsp. `vector < TreeNode* >`). Sie benötigen die Knoten des Suchpfades, um mögliche Rotationen durchzuführen (s. Abschnitt 3).
2. Erzeugen Sie einen neuen Knoten mit den entsprechenden einzufügenden Daten und fügen den neuen Knoten als roten Knoten als Nachfolger des Blattes mit entsprechender Suchrichtung ein (s. Abb. 4 dieser Aufzählung).
3. Untersuchen Sie nun die Knoten entlang des Suchpfades vom eingefügten Blattknoten bis zur Wurzel (Bottom-Up), ob zwei rote Knoten aufeinanderfolgen. Ist das der Fall, müssen die notwendigen Rotationen (s. 4 mögliche Fälle) durchgeführt werden. Sie müssen feststellen, in welcher Richtung rotiert werden muss. Beachten Sie, dass ggfls. die Knoten nach der Rotation umgefärbt werden müssen. Die Abbildungen 5 und 6 verdeutlichen hier eine Doppelrotation. Das Ergebnis nach dem Einfügen kann auch als gültiger 234-Baum interpretiert werden (s. Abb. 7).

- **printLevelOrder()**

Diese Methode gibt den Rot-Schwarz-Baum als 2-3-4-Baum und als Binärbaum in Levelorder-Reihenfolge auf der Konsole aus. Abbildung 7 zeigt links das Ergebnis nach dem Einfügen von Knoten 26 in den Rot-Schwarz-Baum und rechts das zu interpretierende Ergebnis als 2-3-4-Baum. Die entsprechende Ausgabe in Levelorder visualisiert nur die schwarzen Kanten und wäre dann:

Niv. 0: (20,25,30)

Niv. 1: (10) (22) (26,28) (40,50)

Bitte geben Sie zuerst den Binärbaum in Levelorder aus und dann die niveauweise Ausgabe der 2-, 3- und 4-er-Knoten (2-3-4-Baum), die sich aus dem Rot-Schwarz-Baum ergeben. Es genügt hierbei als Schlüssel für die Knoten nur die NodePosID zu nutzen.

Hinweis: Nutzen Sie für die Ausgabe des Rot-Schwarz-Baumes als 2-3-4-Baum-Levelorder zwei Queues. In der einen Queue speichern Sie nur die schwarzen Knoten und in der anderen Queue entsprechend zu dem schwarzen Knoten das Niveau des Knoten im 2-3-4-Baum. Wird ein Knoten aus der Queue entnommen, so entnimmt man auch das dazugehörige Niveau des schwarzen Knotens aus der anderen Queue. Hat der schwarze Knoten rote Nachfolgeknoten, so gehören diese zum schwarzen Knoten (beim 3er oder 4er Knoten bzw. schwarzer Knoten mit 1 roten Nachfolger oder schwarzer Knoten mit 2 roten Nachfolgern) und müssen entsprechend mit ausgegeben werden. Die Nachfolger der roten Knoten sollten dann wieder schwarz sein. Diese nachfolgenden schwarzen Knoten (in der Anzahl 0 bis 4) müssen dann wieder in die Queues mit ihrem entsprechenden 2-3-4-Baum-Niveau gespeichert werden (Erklärung: siehe Video zum Praktikum).

- **rotateTreeRight()/rotateTreeLeft()**

Die beiden übergebenen Knoten werden nach rechts oder links rotiert. Führt man diese Methoden hintereinander aus, erhält man eine Doppelrotation. Geben Sie jeweils auf der Konsole aus, welche Rotation mit welchen Knoten durchgeführt wurden. Beispiel: LR(20,25) oder RR(30,25).

Hinweis: Überlegen Sie in welcher Methode das Umfärben der rotierten Knoten sinnvoll ist im Bezug auf Einzel- und Doppelrotationen.

Testläufe:

```
1 ===== // Beispiel: Menü der Anwendung
2 Person Analyzer v19.84, by George Orwell
3 1) Datensatz einfüegen, manuell
4 2) Datensatz einfüegen, CSV Datei
5 3) Datensatz löschen
6 4) Suchen
7 5) Ausgabe in Preorder
8 6) Ausgabe in Levelorder
9
10 ?> 1 // Beispiel: manuelles Hinzufügen eines Datensatzes
11
12 + Bitte geben Sie die den Datensatz ein
13 Name ?> Mustermann
14 Alter ?> 1
15 Einkommen ?> 1000.00
16 PLZ ?> 1
```

```
17 + Ihr Datensatz wurde eingefügt
18
19 [...] // Hier Eingabe weiterer Datensätze
20
21 ?> 6 // Beispiel: Anzeigen eines Trees in Levelorder Einträgen
22
23 Ausgabe in Levelorder als binärer Suchbaum:
24
25 ID | Name | Alter | Einkommen | PLZ | Pos | Red
26 ---+---+---+---+---+---+---
27 1 | Ritter | 1 | 2000 | 1 | 2002 | 0
28 4 | Schmitt | 1 | 500 | 2 | 503 | 1
29 2 | Kaiser | 1 | 3000 | 1 | 3002 | 0
30 3 | Hans | 1 | 500 | 1 | 502 | 0
31 0 | Mustermann | 1 | 1000 | 1 | 1002 | 0
32 5 | Schmitz | 1 | 400 | 2 | 403 | 1
33
34 Ausgabe in Levelorder als 234-Baum:
35 Niv. 0: (503, 2002)
36 Niv. 1: (403, 502) (1002) (3002)
```

3.2 Lösungshinweise

Beachten Sie ALLE Lösungshinweise und verstehen Sie zunächst wie ein Rot-Schwarz-Baum funktioniert.

3.2.1 Rot-Schwarz-Baum Kriterien

Die Kriterien eines Rot-Schwarz-Baumes müssen zu jeder Zeit alle erfüllt sein!

1. Jeder Knoten ist entweder rot oder schwarz.
2. Jeder neu einzufügende Blattknoten ist rot.
3. Die Kinder von einem roten Knoten sind schwarz.
4. Es gibt keine zwei aufeinanderfolgende rote Knoten.
5. Kriterium für Ausgeglichenheit:

Für jeden Knoten k gilt: Jeder Pfad von k zu einem Blatt enthält die gleiche Anzahl schwarzer Knoten.

6. Die Wurzel ist immer schwarz.

3.2.2 Top-Down Einfügen

Beim Top-Down Einfügen werden alle schwarzen Knoten mit 2 roten Nachfolgern umgefärbt. In Abbildung 8 wird dies für den Knoten n mit seinen Kindern k und p durchgeführt.

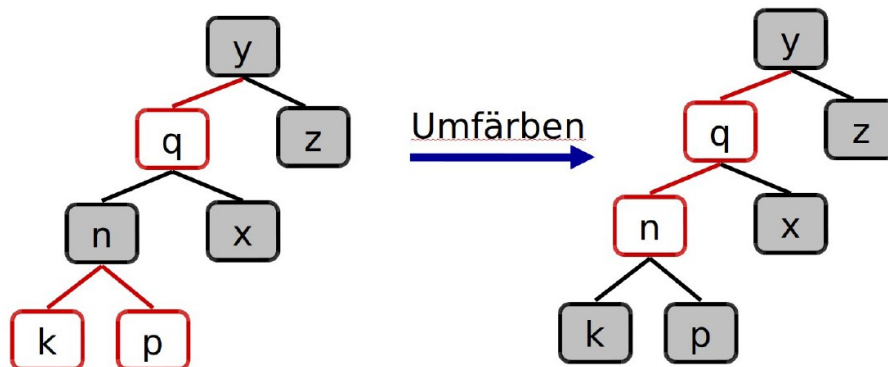


Abbildung 8: Umfärben von Knoten (k, n, p)

Dies kann es zu einer Verletzung des 4. Kriteriums von Rot-Schwarz-Bäumen kommen. Dies muss nach dem Umfärben geprüft und ggf. durch Rotationen behoben werden. Abbildung 3 zeigt ein weiteres Beispiel für das Umfärben.

3.2.3 Rotationen

Durch Einfügen neuer Knoten oder Umfärben können zwei aufeinander folgende Knoten rot eingefärbt werden. Dies verletzt das 4. Kriterium für einen Rot-Schwarz-Baum und der Baum muss durch Rotationen ausgeglichen werden. Möglich sind Rechts-, Links- oder Doppelrotationen (rechts-links oder links-rechts).

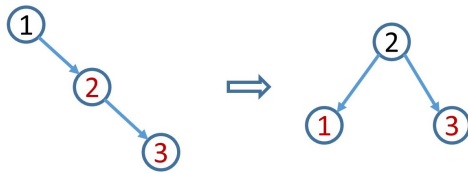
Abbildung 9 zeigt die vier möglichen Rotationen und das jeweilige Ergebnis im Überblick.

Grundsätzlich kann man sich merken:

- Die Richtungen entlang des Suchpfades bis zur Einfügeposition, in der die Knoten verbunden sind, geben die Rotationen vor. Damit ist die Verbindung vom Parent zum rechten oder linken Nachfolger gemeint.
- Zeigen beide aufeinanderfolgende rote Knoten in die gleiche Richtung, so reicht eine einfache Rotation.
- Zeigen die aufeinanderfolgenden roten Knoten in verschiedene Richtungen, so muss mit einer ersten Rotation der untere Knoten an den oberen angeglichen und danach mit einer zweiten Rotation der Baum ausgeglichen werden.

Abbildung 10 zeigt die Knoten 2 und 4 in rot hintereinander. Da Sie nicht aus der gleichen Richtung kommen, muss zunächst eine Linksrotation der Knoten 2 und 4 durchgeführt werden.

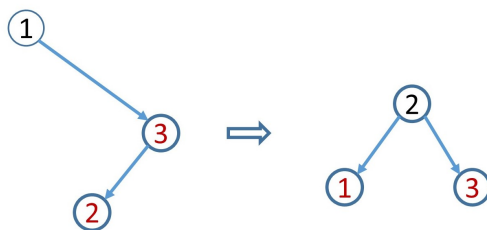
Linksrotation:



Rechtsrotation:



Rechts-Linksrotation:



Links-Rechtsrotation:

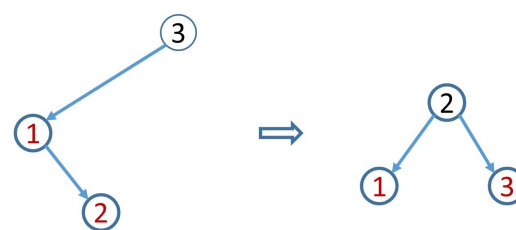


Abbildung 9: Alle vier möglichen Rotationen im Rot-Schwarz-Baum.

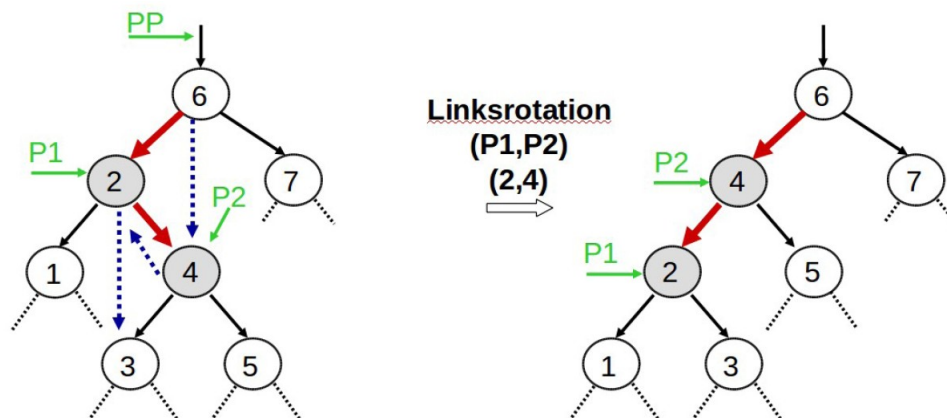


Abbildung 10: Linksrotation der Knoten 2 und 4

Im Anschluss daran wird die zweite Rotation ausgeführt, wie in Abbildung 11 zu sehen. Sie zeigt eine Rechtsrotation der Knoten 6 und 4. Nach dieser Rotation sind die Knoten 2 und 6 rot und die Wurzel 4 ist schwarz.

Eine Doppelrotation setzt sich aus diesen beiden Rotationen zusammen. Die Abbildungen 10 und 11 ergeben eine Doppelrotation in links-rechts Reihenfolge. Daher können Sie in Ihrem Programm die beiden Rotationen hintereinander ausführen um eine Doppelrotation zu erzeugen.

Achtung: Kontrollieren Sie das richtige Einfärben nach Einzel- und Doppelrotationen!

3.2.4 Alte Methoden

Neben den neu zu implementierenden Methoden müssen Sie auch in einigen alten Methoden Anpassungen vornehmen. Folgendes muss beachtet werden:

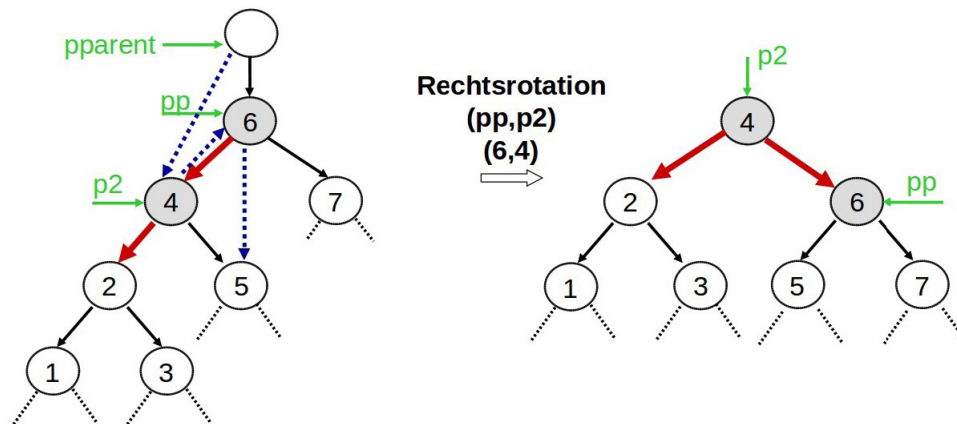


Abbildung 11: Rechtsrotation der Knoten 6 und 4

- Ihre **Einfügemethode muss iterativ programmiert** werden. Haben Sie eine rekursive Einfügemethode verwendet, passen Sie bitte ihre Implementierung an!
- Beim Einfügen wird das Red-Flag des neuen Knotens grundsätzlich auf *true* gesetzt, bei der Umwandlung eines 4er-Knoten kann es zu Umfärbungen kommen, so daß entlang des Suchpfades 2 rote Knoten aufeinander folgen. Der Baum muss auf das Rot-Schwarz-Kriterium "niemals 2 rote Knoten hintereinander" geprüft werden.
- Das Löschen von Knoten wird nicht angepasst. Da es in einem Rot-Schwarz-Baum nicht trivial ist, wird dies hier nicht weiter verwendet oder angepasst.
- In der Suche brauchen Sie keine Anpassungen vornehmen.
- Die Unit-Tests sind für diese Aufgabe angepasst worden, ersetzen Sie die Datei durch die neue von Ilias.